Kalman Filter for Noise Reduction in Aerial Vehicles using Echoic Flow

Andrew Palo – Dr. Inder Gupta

Undergraduate Thesis – Spring 2020

Department of Electrical and Computer Engineering

The Ohio State University

Abstract - Echolocation is a natural phenomenon observed in bats that allows them to navigate complex, dim environments with enough precision to capture insects in midair. Echolocation is driven by the underlying process of echoic flow, which can be broken down into a ratio of the distance from a target to the velocity towards it. This ratio produces a parameter τ representing the time to collision, and controlling it allows for highly efficient and consistent movement. When a quadcopter uses echoic flow to descend to a target, measurements from the ultrasonic range sensor exhibit noise. Furthermore, the use of first order derivatives to calculate the echoic flow parameters results in an even greater magnitude of noise. The implementation of an optimal Kalman filter to smooth measurements allows for more accurate and precise tracking, ultimately recreating the high efficiency and consistency of echolocation tracking techniques found in nature. Kalman filter parameters were tested in realistic simulations of the quadcopter's descent. These tests determined an optimal Kalman filter for the system. The Kalman filter's effect on an accurate echoic flow descent was then tested against that of other filtering methods. Of the filtering methods tested, Kalman filtering best allowed the quadcopter to control its echoic flow descent in a precise and consistent manner. In this presentation, the test methodology and results of the various tests are presented.

Acknowledgements

I want to thank Dr. Graeme Smith and Justin Kuric for helping to get this project off the ground and for making this a valuable learning experience. I also want to thank Dr. Inder Gupta for taking time out of his busy day to serve as this project's advisor, and for doing a great job helping me throughout the process.

Table of Contents

List of Figures and Tables

I. Introduction

Quadcopters and other unmanned aerial systems are used in various applications all around the world. The complexity of the tasks required from these quadcopters is only increasing with time. Examples of these complex tasks include navigating around obstacles such as power lines or trees, as well as autonomously docking at charging stations. These tasks prove to be a large challenge, but luckily a solution to this problem already exists in nature. In this research, the biological concept of echoic flow is used to control a quadcopter's descent. This research plans to accomplish the task of accurately smoothing measurement noise, allowing for echoic flow to precisely control the quadcopter's landing. A more accurate noise filter would also help by decreasing the spread of landing paths taken by the quadcopter, making landings more consistent.

Echoic flow is a biological concept that is used by bats to control their rate of approach and can be characterized as a specialization of General Tau Theory as described by the researchers in [1]. Bats use acoustic flow fields to internally calculate a "time to collision" parameter, τ, which allows them to compute complicated flight paths towards targets or even around obstacles. By keeping the rate of change of τ at a constant value, echoic flow can be used to control an object's motion. This is because an object exhibiting echoic flow follows a curve with an order of the inverse of the rate of change of τ.

With the existing approach, when a quadcopter replicates echoic flow, there are serious shortcomings. First, ultrasonic range sensor measurements exhibit noise. This noise, while not significant, still negatively impacts the precise requirements of the echoic flow algorithm. Furthermore, the use of first order derivatives to calculate echoic flow parameters exacerbates

the noise related error in echoic flow-based control methods. For example, the actual average velocity of the system may be a very small negative value, but noisy measurements could find the average velocity to be a very large positive value. This keeps the quadcopter from performing a precise echoic flow descent. The problem of mitigating measurement noise in echoic flow is extremely significant and important to solve, as quadcopters and drones will continue to play a large role in future technological advancements, and their movement accuracy is paramount in improving them.

A significant amount of research has already been done in the area. Echoic flow was found to demonstrate vast potential for advancing the capability of man-made systems, as seen in [2]. In [3], however, the effect of range and range rate measurement accuracy on echoic flow is discussed. The writers suggest that a filtering algorithm be used to make the measurements more accurate, therefore making the echoic flow itself more accurate. [4] explores a potential filtering algorithm for use in an echoic flow approach. In this application of echoic flow, a quadcopter uses echoic flow to smoothly descend and land. This was accomplished by performing a least squares regression in order to smooth measurement noise. This solution has drawbacks, however. Ideally, the quadcopter would follow the same path for every landing if the rate of change of τ was the same. However, the least squares regression filter assumed that the quadcopter's motion could always be described as a polynomial. This is not always an accurate assumption, and as a result, this filtering technique had an increased spread of landing paths taken by the quadcopter.

One specific key fact about the measurement accuracy problem is that using a least squares regression for the estimation of ranges is not optimal. Another important fact about the problem at-hand is that Kalman Filters significantly reduce noise by estimating a joint probability distribution over many variables and is considered the optimal solution for mitigating

measurement noise by [3]. This research proposes to implement an optimal Kalman filter as a measurement smoother for the echoic flow algorithm discussed in [4].

In order to implement an optimal Kalman filter, the existing system created in [4] had to be refactored. This system was written in JavaScript, a programming language with little to no matrix calculation support, which is required for the implementation of a Kalman filter. The existing codebase was refactored to Python, as the matrix calculation support in Python is far more extensive. Since the system ran on a new codebase, new calibrations had to be made. This includes acoustic range sensor calibrations as well as velocity command calibrations. Once calibrations were complete, experimental trials began. The Kalman filter was tested against a quadratic regression filter at a $\dot{\tau}$ value of 0.5, as [4] found that this was most effective in producing an ideal echoic flow curve. A control of no filter was also tested. Lastly, these same factors were tested in a realistic simulated environment in order to increase sample size.

The major obstacle encountered in this research was accounting for the quadcopter's outdated systems. This includes long charging time and the presence of bugs causing its internal program to crash. These problems made it difficult to achieve a large sample size based on quadcopter trials alone, so most of this research was spent creating and performing an accurate Python simulation of the quadcopter's echoic flow descent. The type of quadcopter used in this research is outdated and therefore accurate simulations of the quadcopter's motion are more indicative of engineering this method for use in real-life situations.

By the time this research had concluded, many things had been learned. One important note is that the measurement noise exhibited by the quadcopter's acoustic range sensor was Gaussian, and because of this, the Kalman filter would be the optimal algorithm for estimating the track of the object. This was experimentally proven to be the case, as the Kalman filter

outperformed the control unfiltered descents as well as the quadratic regression filtered descents. Not only did the median root mean squared error drop significantly, but the spread of the root mean squared error values also significantly dropped. The main lesson from this research was derived from this evidence: the Kalman filter decreased the error and error spread of landing paths taken by the quadcopter when compared to the quadratic regression filter. This means that an echoic flow approach coupled with a Kalman filtering algorithm could be extremely useful for accurately avoiding obstacles and following targets.

In section two of this paper, an overview of echoic flow is given. This includes a detailed description of equations and how the selection of parameters affects the movement of an object. Section two also details how a quadcopter would be controlled with echoic flow. In section three of this paper, the basics of a Kalman filter are described, including specific equations and flowcharts explaining how Kalman filters iteratively use a series of measurements and quantified uncertainty to produce a best estimate for the track of an object. Section three also gives a basic Kalman filter simulation to show its basic effectiveness over other types of filters. In section four of this paper, the experimental methodology is explained. Additionally, the acoustic range sensor calibrations and velocity command calibrations are explained in depth. This section also covers the selection of Kalman filter parameters. Section five of this paper details the results of the experimental trials and discusses why there is a need for simulation trials. In section six, the process for achieving accurate quadcopter error for the simulation is explained, as well as the results of the simulations. Lastly, section seven concludes the paper by providing a summary of the paper's findings, describing further applications of Kalman filtered echoic flow, and discussing potential future work in this area.

II. Echoic Flow

2.1 – Echoic Flow Overview

Echoic flow can be characterized as a specialization of General Tau Theory as described in [1]. In echoic flow, the flow field parameter is the ratio of the range measured by a sensor to the rate of change of that range. This can be expressed as τ, the time to collision, as seen in equation 1.

$$\tau = \frac{r}{\dot{r}} \tag{1}$$

In this equation, r represents the range to the target and $\dot{r}$ represents the change in r over time, or in other words, the relative velocity of the object to the target. The derivative of τ can be described as seen in equation 2.

$$\dot{\tau} = 1 - \tau * \frac{\ddot{r}}{\dot{r}} \tag{2}$$

In this equation, $\ddot{r}$ describes the object's relative acceleration to the target. By selecting a constant value of $\dot{\tau}$ between zero and one, the object will slow down as it approaches the target, allowing motion to be controlled. The application of echoic flow described in [4] found that a $\dot{\tau}$ value of 0.50 was most effective in reproducing an ideal echoic flow motion in experimental trials of a quadcopter's descent.

When an object uses echoic flow, its position can be described as a function of time. This is seen in equation 3 and is given in [2].

$$r = r_0 * (1 + \dot{\tau} * \frac{r_0}{\dot{r}_0} * t)^{1/\dot{\tau}} \quad (3)$$

In this equation, r_0 describes the starting position and $\dot{r}_0$ describes the starting velocity. Additionally, the equation is raised to the $\frac{1}{\dot{\tau}}$ power. Since the object's approach to the target is inversely proportional to $\dot{\tau}$, the choice of $\dot{\tau}$ directly determines the order of the object's range versus time curve. Figure 1 shows the ideal resulting motion for an object's approach at various constant values of $\dot{\tau}$.

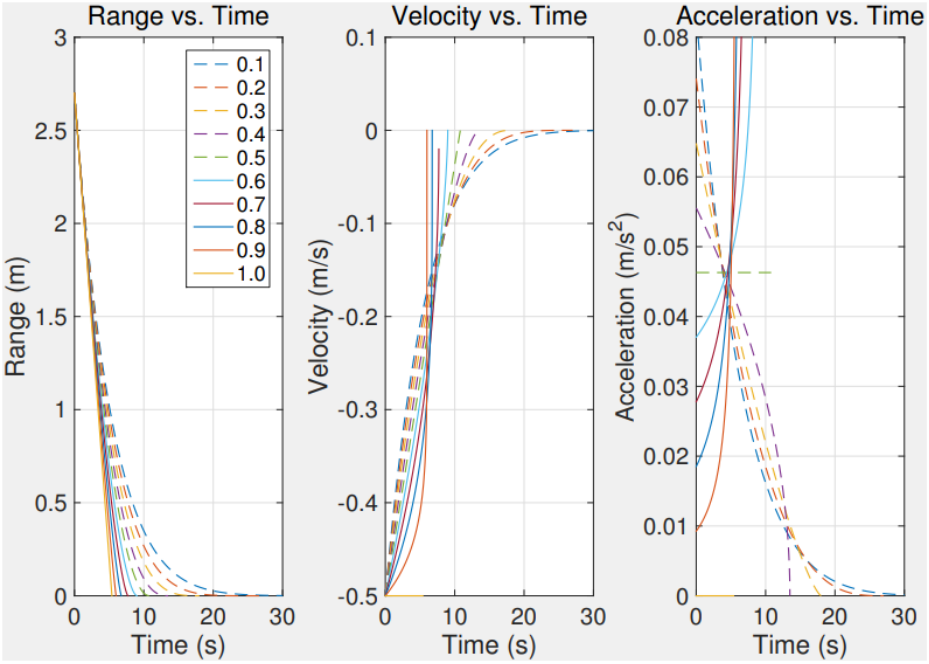

Figure 1: Ideal echoic flow motion for various $\dot{\tau}$ values [4]

2.2 –Echoic Flow Control of a Quadcopter

To control a quadcopter's descent with echoic flow, $\dot{\tau}$ needs to be kept constant throughout its landing. This allows the quadcopter to follow the ideal motion described in Figure

1. To keep $\dot{\tau}$ constant throughout the quadcopter's descent, several steps were taken. First, a measurement was taken from the sensor. After passing this measurement through a smoothing filter, the average velocity was calculated since the last measurement. The filtered measurement and the calculated velocity were then used to calculate τ. The calculated velocity and τ were then used to find the required acceleration of the object. This was accomplished by rearranging equation 2 to solve for $\ddot{r}$. This desired acceleration was required in order to keep $\dot{\tau}$ constant. Equation 4 describes the kinematic motion of an object.

$$v_{n+1} = v_n + a_{n+1} * \Delta t \quad (4)$$

This equation was used to find the velocity needed to achieve the desired acceleration. Finally, the quadcopter would set its velocity to this desired value, and the process would repeat after making a new measurement. This echoic flow control process can be seen outlined in Figure 2.

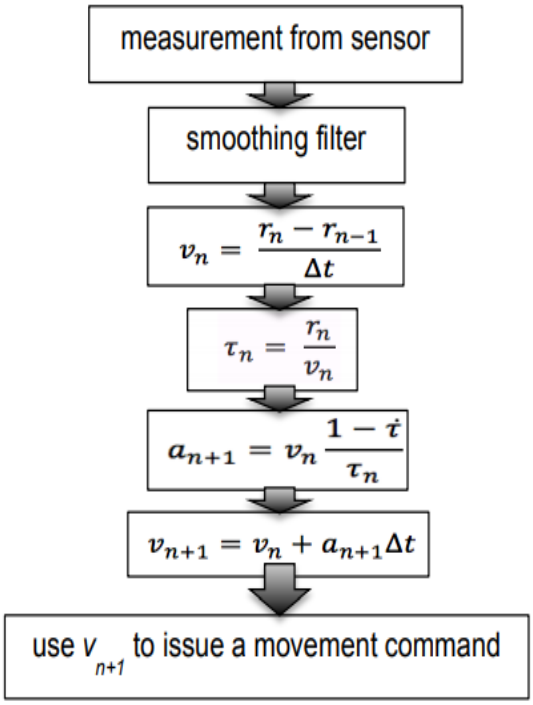

Figure 2: Echoic flow control process of quadcopter [4]

When the natural phenomenon of echoic flow is replicated by a quadcopter, sensor measurements are noisy, as can be seen in Figure 3. This error is only exacerbated when the velocity is calculated in the third step of Figure 2. This causes echoic flow calculations to be erroneous and limits their utility in guidance and control activities. This measurement and calculated error would be best mitigated by a Kalman filter, as this would provide a true estimate of the quadcopter's range.

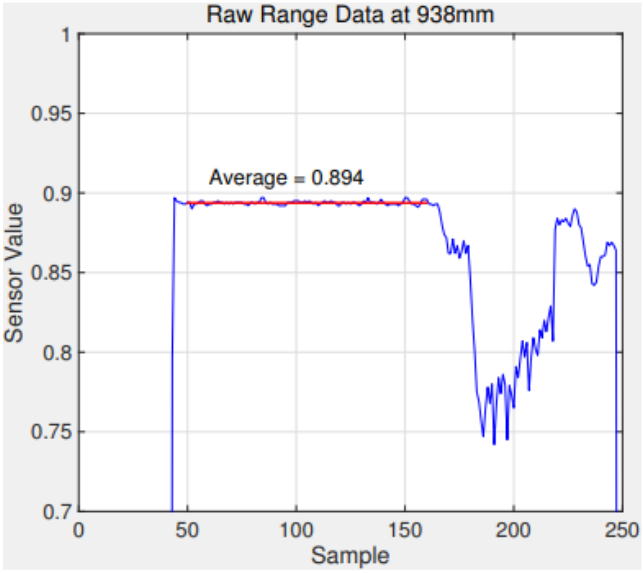

Figure 3: Noisy range measurements at a fixed height [4]

III. Kalman Filtering

3.1 – Kalman Filter Overview

When there is uncertain information about a dynamic system, a Kalman filter can be used to make an educated guess about the system's next state. In this case, the system is a quadcopter descending to the ground with a state x_k, which is just a position α and a velocity v. This will be represented as the matrix in equation 5.

$$x = \begin{bmatrix} \alpha \\ v \end{bmatrix} \quad (5)$$

When a measurement is taken and velocity is calculated, there is no way to know that these are the accurate and precise values. Therefore, every state has uncertainty associated with

it. This uncertainty is described by a covariance matrix, P_k, which can be seen in equation 6. The larger the initial value for P, the more the Kalman filter ignores measurement updates. The smaller the initial value for P, the more the Kalman filter trusts measurement updates. According to [8], the effect of P_0's selection is typically not significant, and it is often initialized to the identity matrix for simplicity.

$$P = \begin{bmatrix} \Sigma_{\alpha\alpha} & \Sigma_{\alpha v} \\ \Sigma_{v\alpha} & \Sigma_{vv} \end{bmatrix} \quad (6)$$

The first step of the Kalman filter is to predict the actual state and the actual uncertainty based on the previous data. Since there are no known external influences, the new state can be described as $x_k = F * x_{k-1}$ where F is described in equation 7.

$$F = \begin{bmatrix} 1 & \Delta t \\ 0 & 1 \end{bmatrix} \quad (7)$$

F is a matrix of kinematic coefficients describing the movement of the quadcopter. While there are no known external influences, there are new unknown influences. This is modeled as process noise covariance Q. In order to specify Q, the quadcopter's motor model and potential physical disturbances must be known. When these models are unknown, Q can be described as white noise with an intuitively determined variance. A larger variance equates to less trust in the state equations, giving more weight to the measurement. The new uncertainty can be described as seen in equation 8.

$$P_k = F * P_{k-1} * \text{transpose}(F) + Q \quad (8)$$

Once the predicted state and uncertainty are found, they must be updated according to a new measurement. This is achieved by combining the two Gaussians of the predicted state and the measurement. This means that for a Kalman filter to be an optimal filter, the measurement noise must have a Gaussian distribution. The updated state has an equation described in equation 9, where K is the Kalman gain, z is the new sensor measurement for the current timestep, and H is the measurement matrix. H = [1 0] because only range is being measured. The updated uncertainty can then be modeled as equation 10.

$$x_{k+1} = x_k + K * (z - H * x_k) \quad (9)$$

$$P_{k+1} = P_k - K * H * P_k \quad (10)$$

When the two Gaussians of the current predicted state and the measurement are combined to get the updated state, a weight is given to the measurement based on uncertainty and measurement noise. This weight is called the Kalman gain and is given by equation 8. R is the measurement noise variance, which describes the spread of the measurement noise of the sensors. If R is large, then the updated state is adjusted primarily based on the predicted state. If R is small, then the updated state is adjusted primarily based on the measurement state.

$$K = \frac{\begin{bmatrix} \Sigma_{\alpha\alpha} \\ \Sigma_{v\alpha} \end{bmatrix}}{\Sigma_{\alpha\alpha} + R} \quad (8)$$

Once the state and uncertainty are updated, these output values can be used to continue with echoic flow, and the entire process loops back on itself, as seen in the flowchart of the Kalman filtering process in Figure 4.

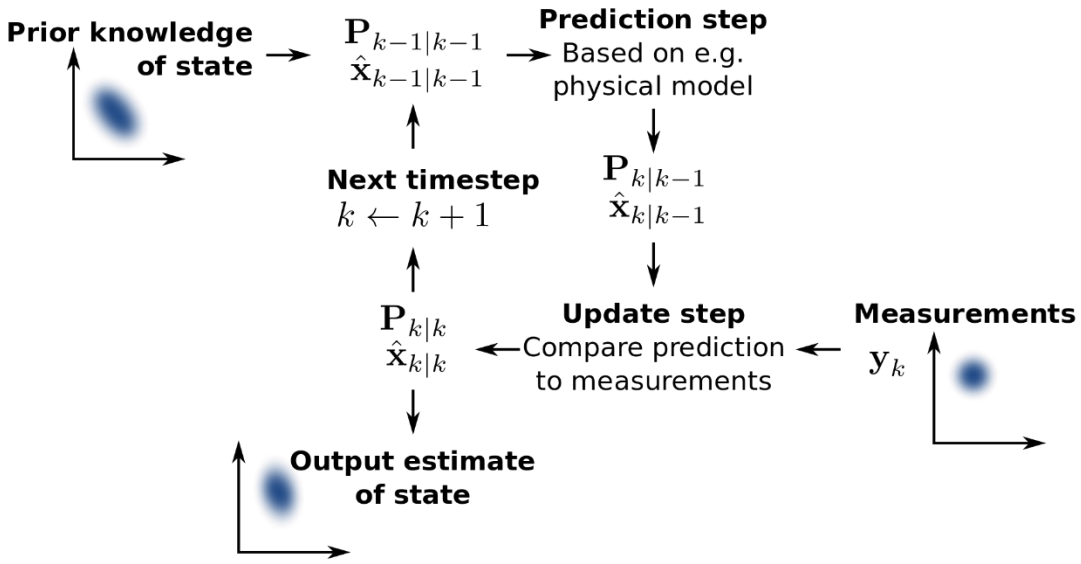

Figure 4: Kalman Filter flowchart [7]

3.2 – Basic Kalman Filter Simulation

To describe the effectiveness of a Kalman filter, a Python simulation of a quadcopter's descent was developed. This basic simulation had the quadcopter starting at a height of 2.0m and descending at a constant rate of 0.3m/s to a height of 0.4m. The quadcopter was given a measurement error normally distributed around 0m with a standard deviation of 0.05m, and the quadcopter's range error and velocity error were plotted over time. Kalman filter parameters

were selected intuitively, as the error is entirely random. P_0 was arbitrarily selected to be the identity matrix for simplicity's sake. R was selected to be 0.0025, as this is the true variance of the measurement noise, and therefore, "would expect the best performance in terms of balancing responsiveness and estimate variance" [6], which is the exact goal of this research. Lastly, Q was described as discrete white noise, as the process noise is completely unknown in this system. In order to visualize the effectiveness of the Kalman filter, it was compared against a quadratic least squares regression filter as well as a control of unfiltered motion. Table 1 describes the range bias and standard deviation of each filtering method for a typical trial, and Figure 5 shows the range error.

	Bias (m)	**Standard Deviation (m)**
Measured Range	0.0009	0.069
Kalman Filtered Range	-0.0030	0.020
Quadratic Regression Filtered Range	0.0013	0.049

Table 1: Bias and standard deviation of a quadcopter's range error over time for measured range, Kalman filtered range, and quadratic least squares regression filtered range

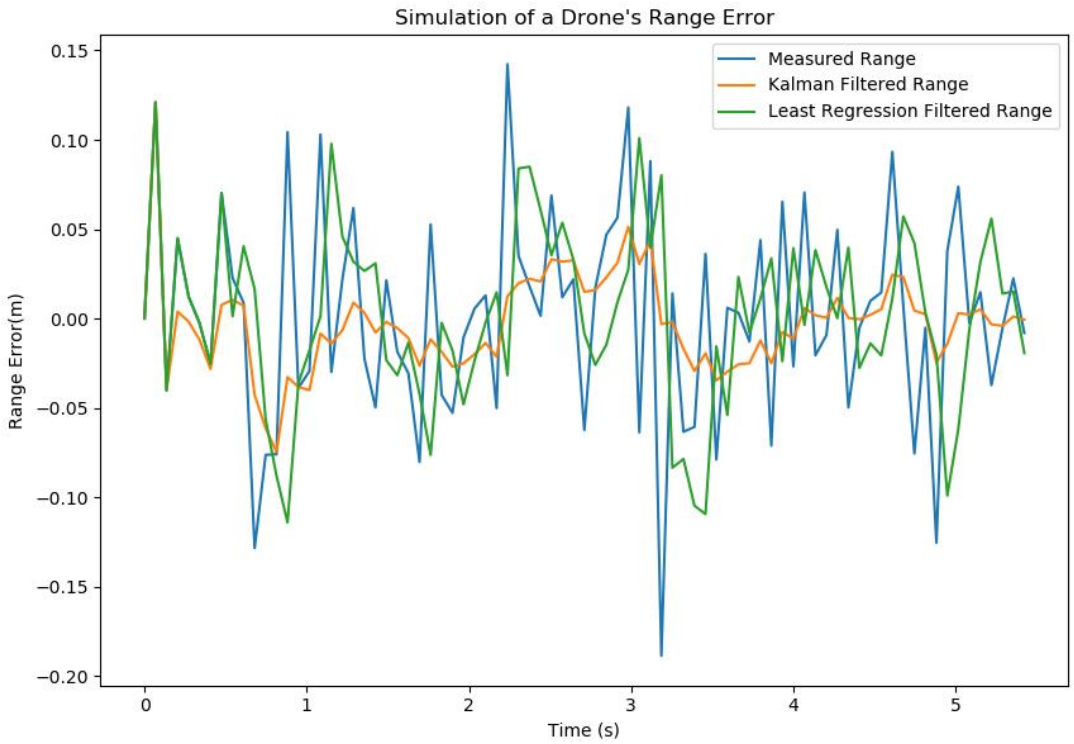

Figure 5: Simulation of a quadcopter's range error over time for measured range, Kalman filtered range, and quadratic least squares regression filtered range

The low bias values indicate that this process is unbiased, as all filtering methods center around a mean error of approximately 0m. The standard deviation values show that the Kalman filter outperforms the least squares regression filter significantly and that having no filter at all yields the most inaccurate range estimates.

The effect of the Kalman filter can be especially noticeable when first order derivatives are calculated. This is because slight errors in range measurement can lead to extreme differences in first order derivative calculations. Table 2 describes the velocity standard deviation of each filtering method for a typical trial, and Figure 6 shows the velocity error.

	Standard Deviation (m/s)
Measured Velocity	1.001
Kalman Filtered Velocity	0.092
Quadratic Regression Filtered Velocity	0.593

Table 2: Standard deviation of a quadcopter's velocity error over time for measured velocity, Kalman filtered velocity, and quadratic least squares regression filtered velocity

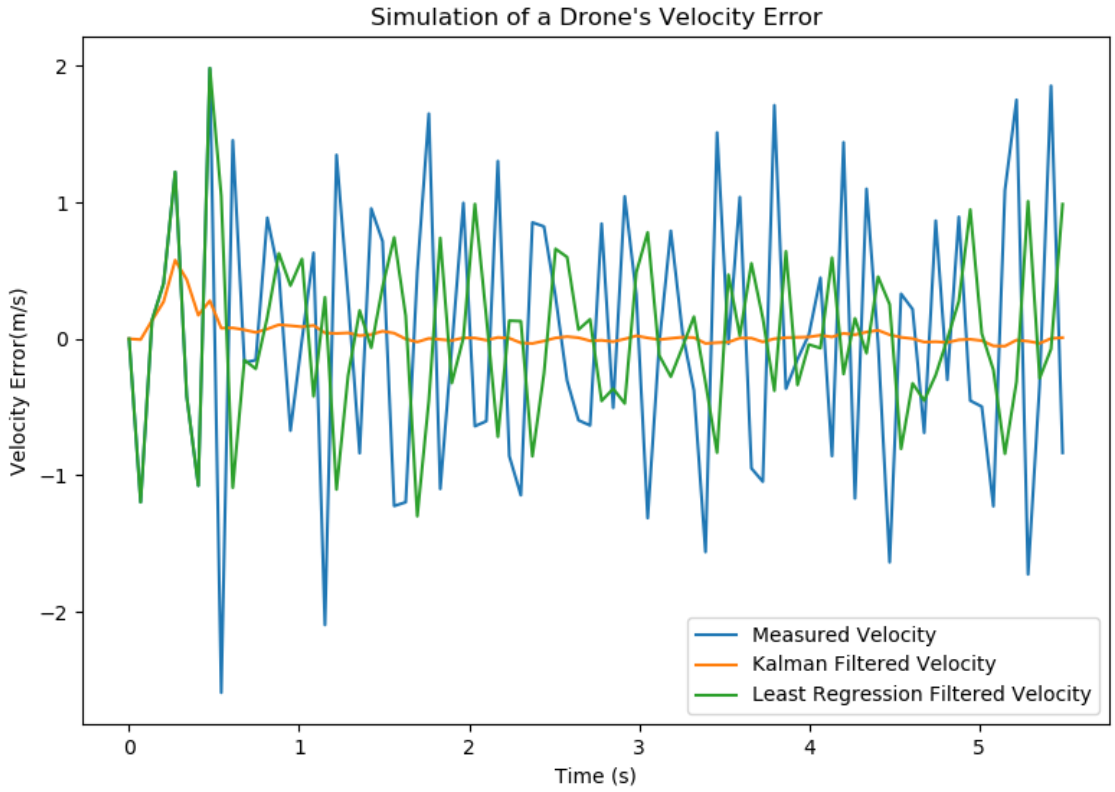

Figure 6: Simulation of a quadcopter's velocity error over time for measured velocity, Kalman filtered velocity, and quadratic least squares regression filtered velocity

Since velocity is calculated rather than measured, bias does not matter. However, the standard deviation values once again show that the Kalman filter outperforms the least squares regression filter significantly and having no filter at all yields the most inaccurate velocity estimates. The velocity error for the Kalman filtered simulation is decreased by more than 85% when compared to the quadratic least squares regression filtered simulation, and by over 90% when compared to the unfiltered simulation. This shows how useful the Kalman filter can be in mitigating measurement noise for both range error and velocity error.

IV. Experimental Methodology, Equipment, and Calibration

4.1 –Experimental Methodology and Equipment

Figure 7 shows the experimental setup as described in [4]. The quadcopter would begin by ascending to a height of 2.0m and then it would approach the ground with an initial velocity of 0.4m/s. After reaching this initial velocity, the quadcopter would use echoic flow to control its descent by iterating through the control process as described in Figure 2. In order to control the landing, $\dot{\tau}$ was kept constant after being selected to be 0.50 at the beginning. This value was used for $\dot{\tau}$ because [4] found it to be the most effective value for a quadratic regression filter, and a Kalman filter's effectiveness is not reliant on the order of an object's motion. $\dot{\tau}$, r_0, and v_0 are the factors that determined the quadcopter's descent, as seen in equation 3. This experiment would be repeated many times with a Kalman filter, a quadratic regression filter, and no filter. The results would then be compared.

Figure 7: Experimental setup [4]

4.2 – Quadcopter and Sensor

The quadcopter used was a Parrot AR Drone 2.0. The Parrot AR Drone 2.0 communicates with a host computer over a Wi-Fi network, and a Python library called PS Drone [5] was used to send movement commands over the network to the quadcopter. In addition, the quadcopter sends information in real time about its height and orientation over the network to the host computer. This quadcopter is equipped with an ultrasonic acoustic range sensor with an operating frequency of 15Hz. The sensor faces downwards and measures the quadcopter's height to three decimal places.

4.3 – Range Sensor Calibration

The acoustic range sensor on the quadcopter is not perfectly accurate. As a result, it needed to be calibrated using a similar process to that described in [4]. The sensor's measurements could be converted into meters by holding the quadcopter above a surface that

was a fixed, known distance away. The measurements were then sent to the computer and saved. This was done for values of exactly 0.500m, 0.600m, 0.700m, 0.800m, and 0.900m. The range sensor's measurements could be converted into true meters by fitting a linear curve to the average measurement value at each known distance. Figure 8 shows a plot of actual distance versus average sensor measurement as well as a corresponding linear conversion equation. The error bars present represent the standard deviation at each data point.

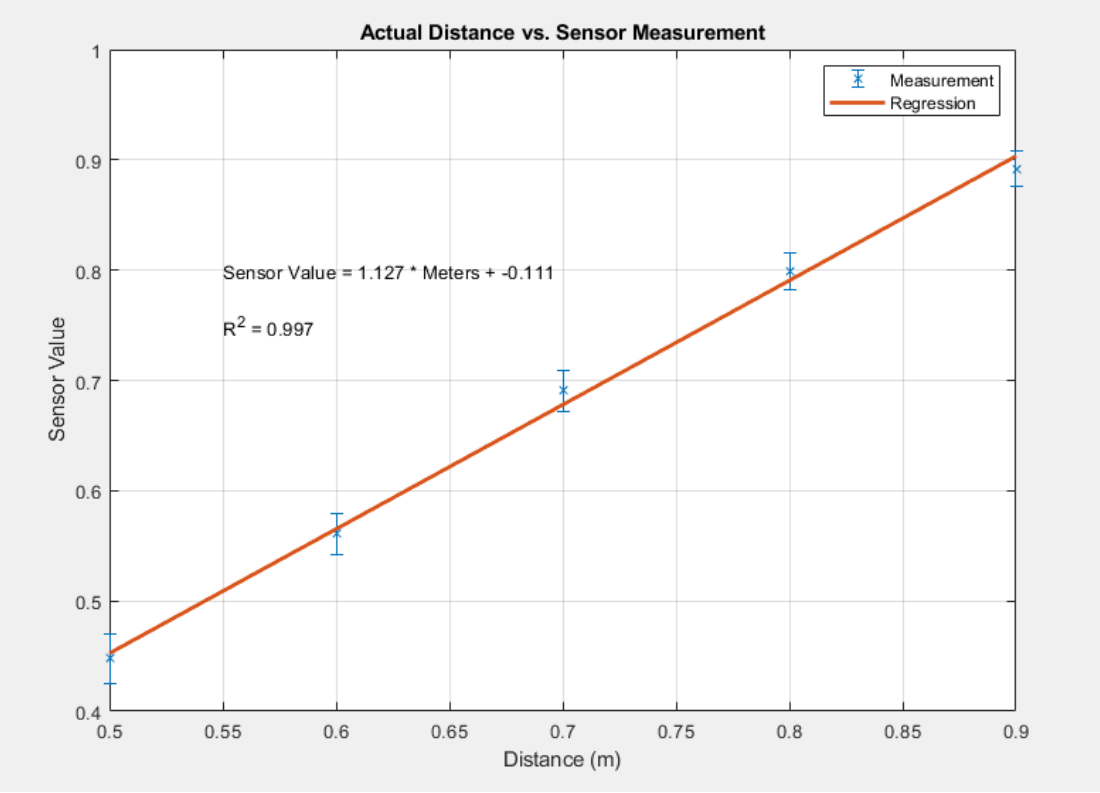

Figure 8: Quadcopter acoustic range sensor calibration

4.4 - Velocity Control Calibration

The PS Drone Python library allows for downwards movement of the quadcopter. In order to initiate this movement from the host computer, a command between 0 and 1 is sent to the quadcopter. This command represents a percentage of the maximum drone speed. Therefore, a command of 0.5 does not necessarily correspond to a downwards velocity of 0.5m/s. Because of this, the command value needed to be converted into actual meters per second in order to control the drone's velocity accurately. In order to do this, the drone flew to a starting height of 2m. From here, a command was given from the host computer instructing the drone to move downwards at a certain percentage of its maximum speed.

A range versus time plot was created from these measurements and the slope of this constant downwards movement was found in m/s. This process was repeated for commands of 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, and 0.8. Commands of 0.9 and 1.0 were omitted because these commands are never sent to the quadcopter throughout the duration of the echoic flow descent. Five trials were done for each of the eight commands, and the average true velocity was found for each. The average true velocities were then plotted against the command values, and a resulting quadratic curve was fit. This plot can be seen in Figure 9, where the error bars represent the standard deviation of the velocity. The inverse of this quadratic equation gives the host computer a methodology for converting the desired velocity to a command.

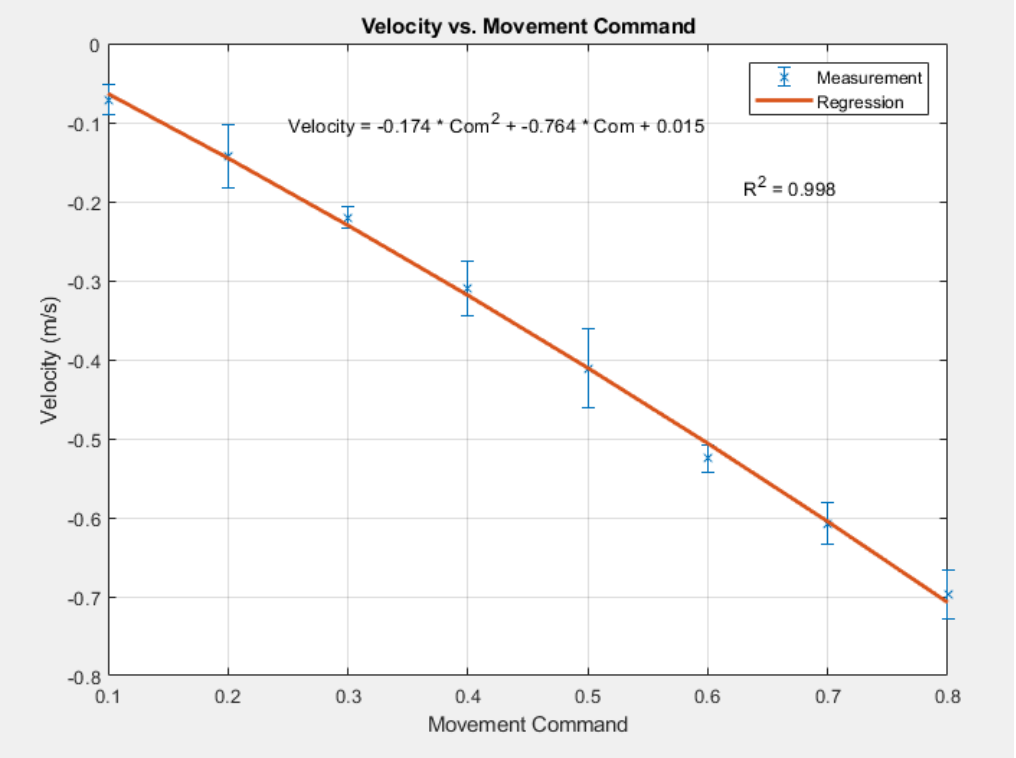

Figure 9: Plot of true velocity vs. quadcopter movement command

4.5 – Kalman Filter Parameter Tuning

Kalman filter parameters were selected using various methods. P_0 was selected to be the identity matrix for simplicity. R, on the other hand, was found experimentally. In order to accurately determine R, accurate measurement noise needed to be determined. It was known that measurement error was dependent on altitude, as the quadcopter produces upward wind drafts when closer to the ground, which affects the range sensor's accuracy [4]. Since the measurement noise was dynamic, R was also dynamic. To experimentally find measurement noise at various heights, the quadcopter was held at a fixed height for 1000 measurements, and the standard deviation of the measurement error was found. An example of one of these runs can be seen for a height of 0.75m in Figure 10. The standard deviation of these measurements was found to be 0.0184m. Therefore, at a height of 75cm, R would equal 0.0184^2, or 0.000339. This was repeated

ten times for each of various altitudes ranging from 0.5m to 2.0m, and a resulting quadratic fit curve was found, as seen in Figure 11.

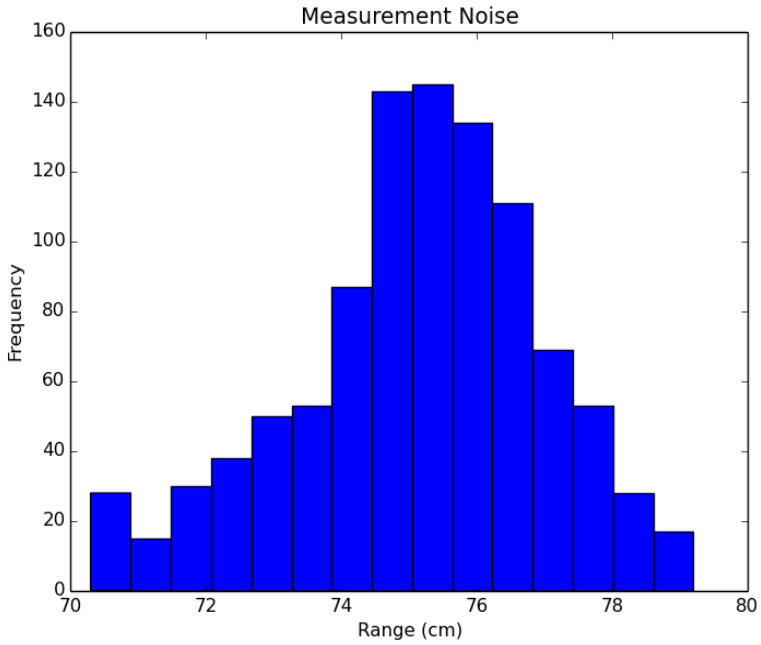

Figure 10: Frequency plot of measured ranges at a fixed height of 0.75m

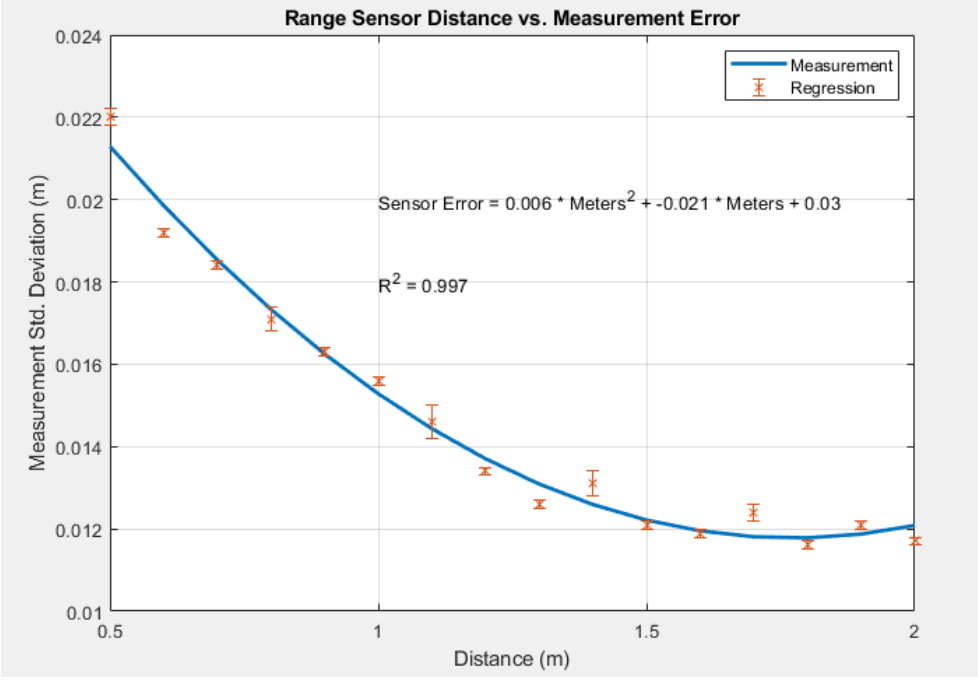

Figure 11: Plot of range sensor distance vs. measurement standard deviation

It is important to note that Figure 10 shows that the measurement noise is Gaussian. This means that the Kalman filter is the optimal filtering algorithm for the system. Lastly, Q was described as discrete white noise, as the quadcopter's motor models are completely unknown in this system. The variance of this white noise was selected experimentally. Initial tests found that white noise variance values between 0 and 0.01 were best for minimizing root mean squared error (RMSE) when compared to an ideal descent. Three Kalman filtered echoic flow descents were done for each of the variance values seen in Table 3. The median of the three runs was then recorded. As can be seen in the shaded portion of Table 3, a white noise variance of 0.005 was the best at minimizing error.

Q White Noise Variance	0.001	0.002	0.003	0.004	0.005	0.006	0.007	0.008	0.009	0.01
Median RMSE (cm)	29.01	34.11	23.73	20.97	18.85	21.38	24.65	30.98	27.12	29.74

Table 3: Median RMSE for different values of Q's white noise variance (N = 3)

V. Echoic Flow Experimental Results

Echoic flow descent trials were then done with the quadcopter. Ten trials were done with a Kalman filter, ten trials were done with a quadratic least squares regression filter, and ten trials were done without a smoothing filter at all. The quadratic least squares regression filter was tuned according to [4], which found a buffer size of 19 to be most effective at minimizing RMSE. Table 4 shows the results of these experimental trials. The median was calculated instead of the mean, as large RMSE values greatly skewed the mean. The standard deviation was also

found in order to see each filter's effect on the consistency of low-RMSE simulations. Additionally, the percentage of simulations with a notably low RMSE (< 10 cm) was found.

	Median RMSE (cm)	**RMSE Standard Deviation (cm)**	**Percentage of Simulations with RMSE < 10 cm**
No Filter	67.92	53.08	0%
Quadratic Filter	37.18	46.77	10%
Kalman Filter	23.13	20.21	40%

Table 4: RMSE statistics for experimental echoic flow descent with different filtering algorithms (N = 10)

Table 4 shows that the Kalman filter was the best and most consistent at mitigating noise and allowing the quadcopter to perform an accurate echoic flow descent. Figure 10 shows the best Kalman filtered descent compared against an ideal echoic flow curve.

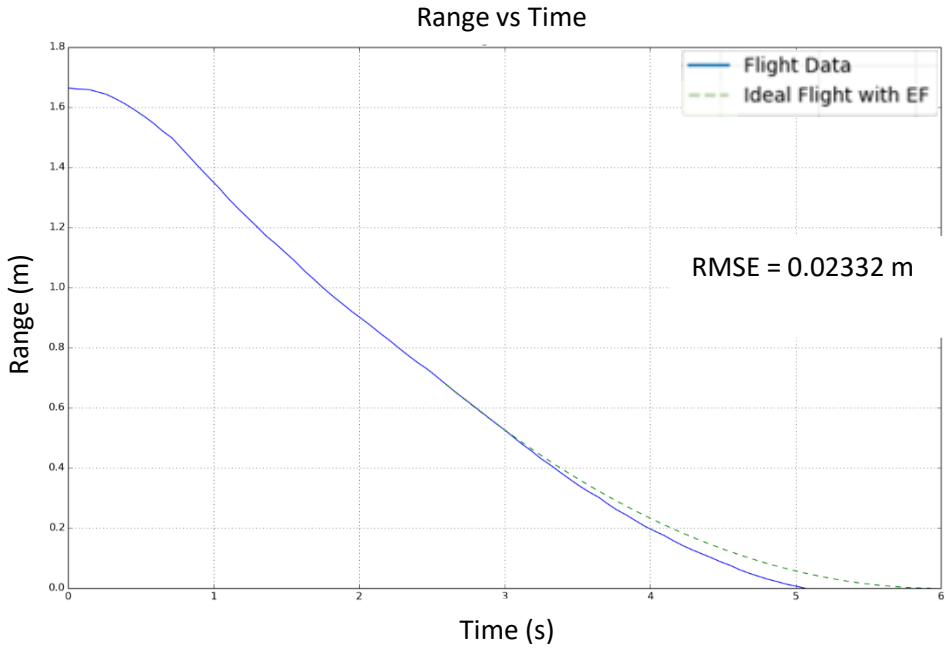

Figure 12: Quadcopter's best actual echoic flow descent with Kalman filter

The actual quadcopter trials required a full battery charge to be most accurate, and the quadcopter's outdated system crashed the program on the host computer often and unpredictably. These factors led to a low sample size for experimental trials. This low sample size coupled with a high error spread meant that a much higher sample size was required. Therefore, accurate simulations were the only way to get a true idea of which filter worked best at mitigating measurement noise.

VI. Echoic Flow Simulation

6.1 – Error Calibration for Simulation

For an accurate echoic flow simulation, accurate measurement error needed to be determined. Accurate measurement error was previously found in section 4.5, when calculating the Kalman filter parameter R. Therefore, when building the simulation, it was known that measurement error was based on the quadcopter's height according to the equation in Figure 11. This equation could be used in the echoic flow simulation to accurately depict measurement noise based on the current altitude. This was done by introducing a random error normally distributed around zero with a standard deviation of based on the equation in Figure 11.

Velocity also had an error associated with it, as the input command would not always correspond perfectly with the desired velocity. This was determined experimentally based on the velocity calibration standard deviation values calculated in Figure 9. These standard deviation values were plotted against their corresponding requested velocity. The error was found to be independent of desired velocity, as any best fit curve had a low correlation ($R^2 < 0.5$). However, the velocity error was still normally distributed around a mean of zero. An example of this can be

seen in Figure 13 at a requested velocity of 0.5m/s. Fifteen trials were done in order to get a better visual of the normal distribution. Therefore, a simple average of velocity error standard deviation was taken across various input commands of 0.1 to 0.8. This average was found to be 0.031m/s. This value was used in the echoic flow simulation to accurately depict velocity error when giving the simulation a velocity command. This was done by introducing a random error normally distributed around zero with a standard deviation of 0.031.

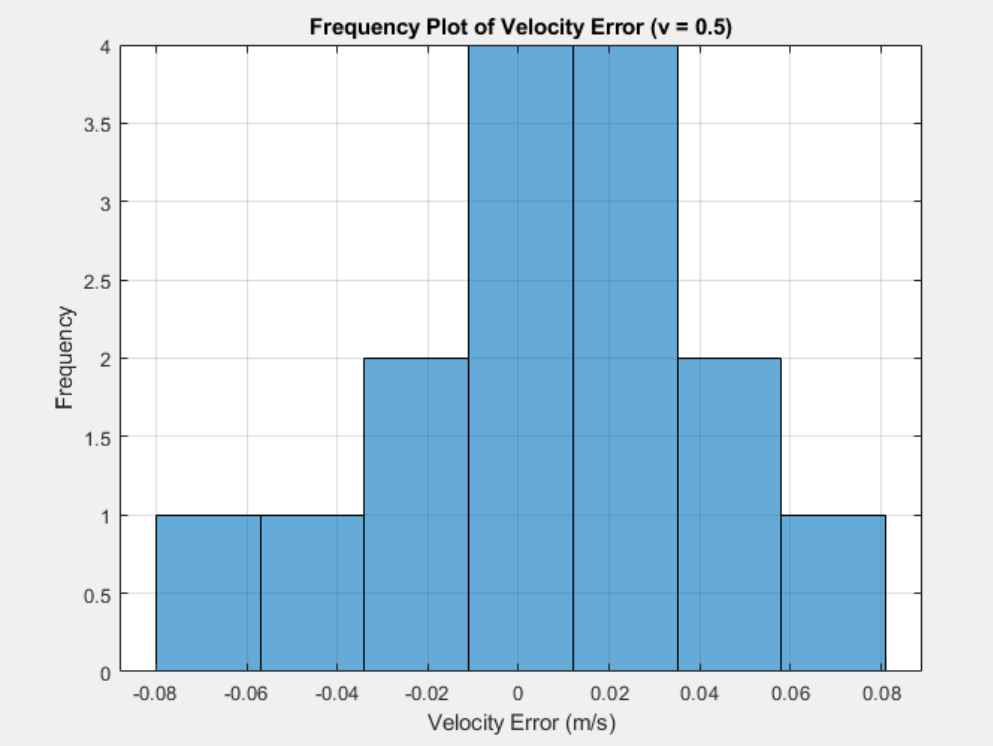

Figure 13: Frequency plot of velocity error at a requested velocity of 0.5m/s

6.2 – Echoic Flow Simulation Methodology

The simulation was created in Python and was based directly off the echoic flow quadcopter descent code. While it did not make use of the PS Drone library, it used simulated

velocity ‘commands’ and 15Hz time delays in order to accurately simulate the quadcopter’s behavior. Figure 14 shows an echoic flow simulation with no error introduced. The RMSE of 0 indicates that without error, the echoic flow descent works ideally. After the simulation error found in section 6.1was introduced, individual simulations were done of each of the three filtering methods. This gives a visualization of how accurate the filters are at maintaining an ideal echoic flow descent. An example of a Kalman filtered simulation can be seen in Figure 15, where the RMSE is very low at only 1.351cm.

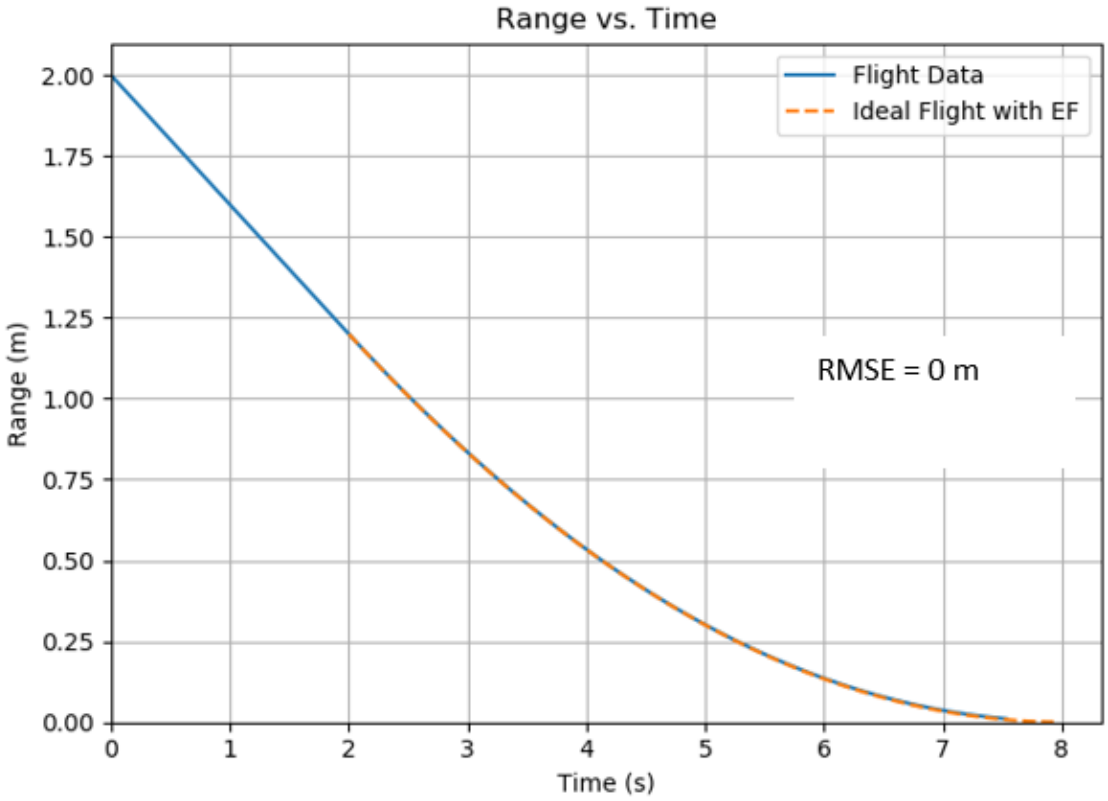

Figure 14: Simulation of echoic flow with no introduced error

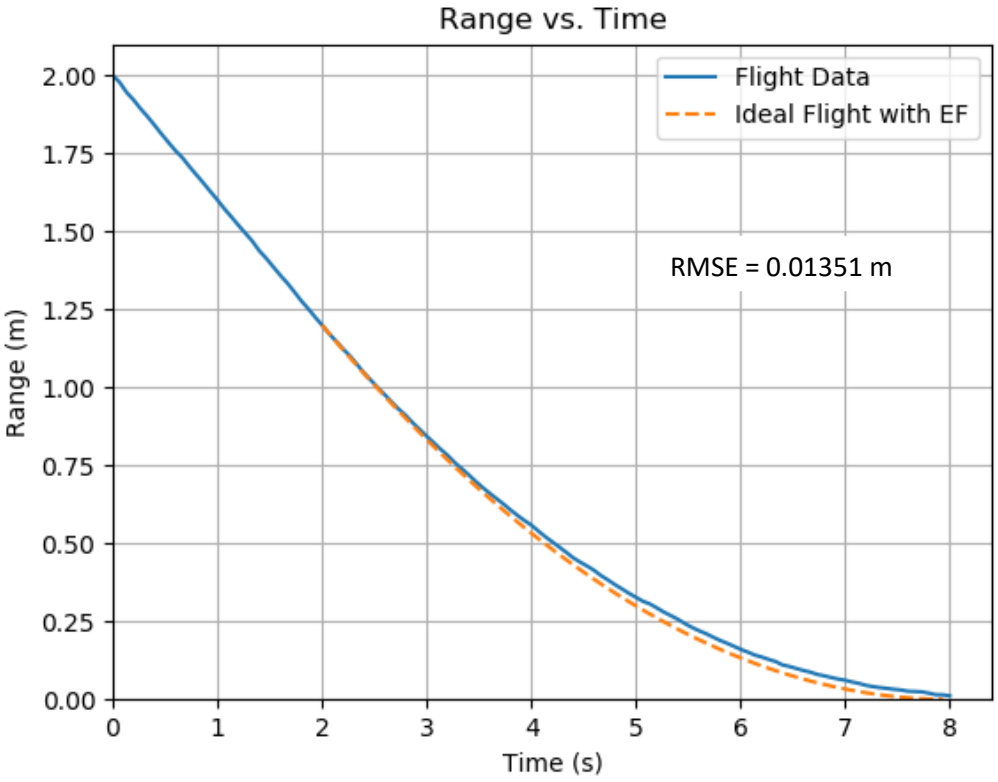

Figure 15: Kalman filtered echoic flow descent simulation with realistic quadcopter error

6.3 – Echoic Flow Simulation Results

After setting up the simulation and testing to ensure that it worked accurately, 1000 descent simulations were done for each filtering method. This large sample size helps to get a better idea of which filter is best for mitigating echoic flow descent error. The results of these simulations can be found in Table 5. The median was calculated instead of the mean, as bad runs with large RMSE values greatly skewed the mean. Additionally, the standard deviation was found in order to see each filter's effect on the consistency of low-RMSE simulations. Lastly, since bad runs are inevitable with so much error being introduced, the percentage of simulations with a notably low RMSE (< 10 cm) was also found.

	Median RMSE (cm)	RMSE Standard Deviation (cm)	Percentage of Simulations with RMSE < 10 cm
No Filter	49.04	75.45	0.4%
Quadratic Regression Filter	22.05	48.14	18.2%
Kalman Filter	14.11	11.97	32.6%

Table 5: RMSE statistics for echoic flow descent simulations with different filtering algorithms (N = 1000)

Table 5 shows how the Kalman filtering algorithm is optimal in mitigating measurement noise and allowing the simulated quadcopter to descend according to the ideal echoic flow curve. Additionally, the Kalman filter increases the consistency of low-RMSE simulations, greatly outperforming the quadratic least squares regression filter and especially outperforming the absence of a filter.

VII. Conclusion

The landing of a quadcopter was controlled using a natural phenomenon called echoic flow. When performing an echoic flow descent with a quadcopter, the measurement noise of the acoustic range sensor causes the descent to differ from the ideal echoic flow curve. The implementation of a Kalman filter to smooth measurements allows for more accurate and precise tracking, ultimately recreating the high efficiency and consistency of echolocation tracking techniques found in nature. The main conclusion of this research is that a Kalman filter works

better to minimize measurement noise and echoic flow error spread when compared to other filtering methods, allowing for an efficient and consistent tracking. Some real-life applications of this method include landing drones in a specific location extremely precisely.

The type of quadcopter used in this research is outdated. The battery life and presence of various fatal bugs caused the sample size of echoic flow trials to be low. The echoic flow descents are not always near ideal, causing the need for a very large sample size. Therefore, accurate simulations of the quadcopter's echoic flow descent are more indicative of engineering this method for use in real-life situations.

In the future, using a different type of battery with a longer life could allow for more trials to be done, and for a better indication of how each filter worked at mitigating measurement noise. Additionally, future work could investigate the effectiveness of $\dot{\tau}$ values other than 0.50 when measurements are Kalman filtered. For this research, 0.50 was used as a basis of comparison, as it was previously found to be the most effective $\dot{\tau}$ value for a quadratic filter. Lastly, future research could look at implementing a Kalman filter on an echoic flow approach algorithm towards a moving object, which more accurately depicts what is seen in nature. It would also have more real-life implications than a simple landing.

References

[1] Lee, D. N. "General Tau Theory: Evolution to Date." Perception vol. 38, pp. 837-58. 2009.

[2] C. J. Baker, G. E. Smith, A. Balleri, M. Holderied, and H. D. Griffiths, “Biomimetic Echolocation With Application to Radar and Sonar Sensing,” Proc. IEEE, vol. 102, no. 4, pp. 447–458, Apr. 2014.

[3] S. A. Alsaif, G. E. Smith, and C. J. Baker, “Echoic flow for target following and approach,” 2016 CIE International Conference on Radar (RADAR), Jan. 2016.

[4] J. Kuric, “Using Echoic Flow for the Guidance and Control of an Unmanned Aerial System,” thesis, 2016.

[5] PS-Drone - Programming a Parrot AR.Drone 2.0 with Python - The Easy Way. [Online]. Available: http://www.playsheep.de/drone/. [Accessed: 17-May-2019].

[6] G. Welch and G. Bishop, “An Introduction to the Kalman Filter - Computer Science,” University of North Carolina at Chapel Hill, 24-Jul-2006. [Online]. Available: http://www.cs.unc.edu/~welch/media/pdf/kalman_intro.pdf. [Accessed: 02-Mar-2020].

[7] “Kalman filter,” Wikipedia, 21-Mar-2020. [Online]. Available: https://en.wikipedia.org/wiki/Kalman_filter. [Accessed: 02-Apr-2020].

[8] M. B. Rhudy, R. A. Salguero, and K. Holappa, “A Kalman Filtering Tutorial for Undergraduate Students,” International Journal of Computer Science & Engineering Survey, vol. 08, no. 01, pp. 01–18, Feb. 2017.